



Week 7  
Review and file  
processing

# >>> Methods

```
1 public class Hello {
2     public static void main(String[] args){
3         hello();
4     }
5     public static void hello(){
6         System.out.println("Hello \"world\"!");
7     }
8 }
9
```

## hello.py

```
1 def hello():
2     print "Hello \"world\"!"
3
4 #main
5 hello()
```

### Structure:

```
def <method name>():
    <statement>
    <statement>
    <statement>
    ...
    <statement>
```

**Note:** Python does not have a main method. However, methods must be defined first. Calls to methods can then be made. This is similar to having a “main” section of code. It is good practice to label this section of code with a comment.

**Second Note:** Remember that whitespace and especially tabs are important!



# >>> Python's For

Unlike Java's for loop, Python's for loop loops over elements in a sequence. To loop over a certain sequence of integers use the range() function. Later we will learn objects that we can use a for loop to go through all of the elements.

## for.py

```
1 for i in range(4): # (end)
2     print i
3
4 for i in range(1,4): # (start,end)
5     print i
6
7 for i in range(2,8,2): # (start,end,step_size)
8     print i
9
10 # for <name> in range([<min>,) <max> [,<step>]):
11 #     <statements>
12
```



# >>> Nested loops

In Python, a lot of the time we can do nested loops in a much more straightforward way using string multiplication.

## Nested.java

```
1 for (int i = 1; i <= 5; i++) {
2     for (int j = 1; j <= (5 - i); j++) {
3         System.out.print(" ");
4     }
5     for (int k = 1; k <= i; k++) {
6         System.out.print(i);
7     }
8     System.out.println();
9 }
```



```
5
44
333
2222
11111
```

## nested1.py

```
1 for i in range(5,0,-1):
2     print " " * (i-1) + str(i)*(6-i)
3
```

## nested2.py

```
1 import sys
2     for i in range(5,0,-1):
3         sys.stdout.write(" " * (i-1))
4         sys.stdout.write(str(i)*(6-i))
5     print
```

# >>> Parameters

Parameters are easy in Python once you know Java's. Simply remove all types from the method header and do the normal conversion.

## PrintSum.java

```
1 public static void printSum(int x, int y) {  
2     System.out.println(x + y);  
3 }  
4
```

## print\_sum.py

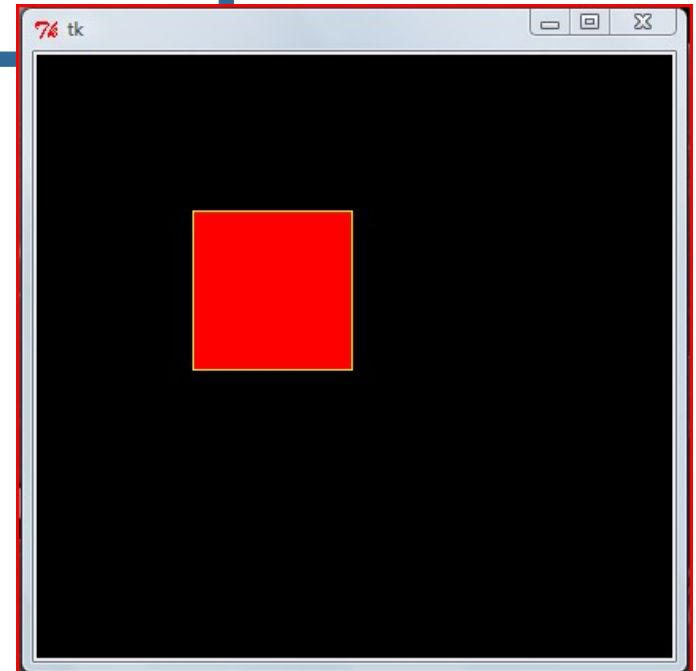
```
1 def print_sum(x, y):  
2     print str(x + y)  
3  
4 sum(2, 3)
```



# >>> Graphics Example

## Python

```
1 from drawingpanel import *
2 panel = DrawingPanel(400,380)
3 g = panel.get_graphics()
4 g["bg"]="black"
5 g.create_rectangle(100, 100, 200, 200, fill="red", outline="yellow")
6 panel.mainloop()
```



# >>> if

Like many things in the transition from Java to Python, curly braces are replaced with colons and whitespace, the parentheses are dropped and &&, || and ! change.

## Translator.java

```
1 // 1 for english
2 // 2 for german
3 int translator = 1;
4 if (translator == 1) {
5     english();
6 } else if (translator == 2) {
7     german();
8 } else {
9     System.out.println("None");
10 }
```



## Java

<  
>  
<=  
>=  
==  
!=  
||  
&&  
!

## python

<  
>  
<=  
>=  
==  
!=  
or  
and  
not

**Notice:** "else if" becomes "elif"

## translator.py

```
1 translator = 1
2 if translator == 1:
3     english()
4 elif translator == 2:
5     german()
6 else:
7     print "None"
```

# >>> input() vs. raw\_input()

There are two ways of getting input. The first is `input()`. It takes in input until enter is hit and then tries to interpret it into python. However, this way only works well for numbers.

```
>>> x = input("yes? ")
yes? y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<string>", line 1, in <module>
NameError: name 'y' is not defined
>>> x = input("yes? ")
yes? 2
>>> print x
2
>>> x = input("num? ")
num? 2.0
>>> print x
```

The second way is to use `raw_input()` which returns the entire line as a string. Once this is done, the string can be split into smaller strings and changed to the desired type.

## inputs.py

```
1 #take a number in
2 x = input("x? ")
3 print x
4
5 #take a sentence to tokenize it
6 sent = raw_input("sentence: ")
7 for w in sent.split(" "):
8     print "word: " + w
9
```





# >>> while

The while loop translates nicely from Java to Python.

## sentinel.py

```
1 sum = 0
2 number = input("Enter a number (-1 to quit)? ")
3
4 while number != -1:
5     sum += number
6     number = input(" Enter a number (-1 to quit)? ")
7
8 print "The total is " + str(sum)
9
```

## Sentinel.java

```
1 Scanner console = new Scanner(System.in);
2 int sum = 0;
3 System.out.print("Enter a number (-1 to quit): ");
4 int number = console.nextInt();
5
6 while (number != -1) {
7     sum = sum + number;
8     System.out.print("Enter a number (-1 to quit): ");
9     number = console.nextInt();
10 }
11
12 System.out.println("The total is " + sum);
```



# >>> random

Just like in Java, python also has random object. Here is an example:

```
>>> from random import *
>>> randint(0,9)
1
>>> randint(0,9)
4
>>> choice(range(10))
7
```

**random.randint(a,b)**

returns an int between a and b inclusive

**random.choice(seq)**

returns a random element of the sequence



# >>> hours worked

Sample output:

```
Employee 1: How many days? 3
Hours? 1
Hours? 2
Hours? 9
Employee 1's total hours = 11
Employee 2: How many days? 2
Hours? 8
Hours? 10
Employee 2's total hours = 16
Total hours for both = 27
```

```
// Marty Stepp, CSE 142, Autumn 2007
// This program reads the hours worked by two employees and
// outputs the total hours for each and for both employees.

import java.util.*; // so that I can use Scanner

public class Hours {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        int totalHours = 0;
        totalHours = totalHours + processEmployee(console, 1);
        totalHours = totalHours + processEmployee(console, 2);
        System.out.println("Total hours for both = " + totalHours);
    }

    // This method reads the hours worked by one employee
    // and returns the total hours.
    public static int processEmployee(Scanner console, int num) {
        System.out.print("Employee " + num + ": How many days? ");
        int days = console.nextInt();

        // cumulative sum for hours worked each day
        int sum = 0;
        for (int i = 1; i <= days; i++) {
            System.out.print("Hours? ");
            int hours = console.nextInt();
            sum += Math.min(8, hours); // cap to 8 hours in one day
        }
        System.out.println("Employee " + num + "'s total hours = " + sum);
        return sum;
    }
}
```



# >>> graphic example - rectangles

```
from random import *
from drawingpanel import *

def drawRandomRect():
    x = randint(0,490)
    y = randint(0,490)
    randomColor = choice(("red", "orange", "yellow", "green", "blue", "purple"))
    size = randint(1,100)

    g.create_rectangle(x, y, x+size, y+size, fill=randomColor)

    return randomColor == "red"

#main
panel = DrawingPanel(500, 500)
g = panel.get_graphics()
reds = 0

while reds < 20:
    if drawRandomRect():
        reds += 1
```



# >>> reading files

File processing in Python is somewhat different than Java. However, its easier.

Opening files:

`open(filename)` ~ defaults to read

`open(filename, "r")` ~ specifies read

`open(filename, "w")` ~ writes to this file

File objects: (we won't really have to use these)

\* `.readlines()` ~ file as a list of lines

\* `.read()` ~ file as a string

\* `.readline(e)` ~ next line as string

## imdb.py

```
1 filename = "imdb.txt"
2
3 f1 = open(filename)
4
5 for line in f1:
6     print line.upper()
7
8 f1.close()
```



# >>> token processing

We are going to do token processing much differently than in Java.

`split()` ~ splits up line between whitespace  
`split([string])` ~ splits up line based on string

iterating over tokens~

for [name] in [line].split([string]):

searching in a line~

if [search] in [line].split([string]):

## imdb2.py

```
1 line1 = "this is a sentence"
2 line2 = "python, java, c, ruby"
3
4 for token in line1.split():
5     print token
6
7 for token in line1.split("s"):
8     print token
9
10 for token in line2.split(","):
11     print token
```



# >>> unpacking a line

To get specific things on a line, we will unpack it like the point tuples. We will learn another way to do this next week, but this is good enough for now!

## unpack.py

```
1 data = "data 3843 9034"  
2  
3 label, data1, data2 = data.split()  
4  
5 data1 = int(data1)  
6 data2 = int(data2)  
7  
8 print label + " " + str(data1 + data2)  
9
```



# >>> movies!

## Input file:

```
1 % 9.1 % 243153 % The Godfather (1972)
2 % 9.1 % 287728 % The Shawshank Redemption (1994)
3 % 9.0 % 139085 % The Godfather: Part II (1974)
4 % 8.9 % 76914 % Buono, il brutto, il cattivo, II (1966)
...
```

## Sample output:

```
Search? pulp
5 247454 8.8 Pulp Fiction (1994)
1 match(es).
```



```
import java.awt.*; // for Graphics
import java.io.*; // for File
import java.util.*; // for Scanner

public class Movies {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner console = new Scanner(System.in);
        System.out.print("Search word? ");
        String searchWord = console.next(); // "Pulp"
        searchWord = searchWord.toLowerCase();

        // look for that word in the file
        int matches = 0;
        Scanner input = new Scanner(new File("imdb.txt"));
        while (input.hasNextLine()) {
            // read a line, see if it matches, print it
            String line = input.nextLine();
            String lineLC = line.toLowerCase();

            int index = lineLC.indexOf(searchWord);
            if (index != -1) {
                // line matches
                matches++;
                Scanner lineScan = new Scanner(line);
                int rank = lineScan.nextInt(); // 5
                double rating = lineScan.nextDouble(); // 8.8
                int votes = lineScan.nextInt(); // 247454

                String title = "";
                while (lineScan.hasNext()) { // print title
                    String token = lineScan.next();
                    title = title + " " + token;
                }

                System.out.println(rank + " " + votes + " " + rating + " " + title);
            }
        }

        System.out.println(matches + " matches.");
    }
}
```